

EEEEEEEEEE	DDDDDD	IIIIII
EEEEEEEEEE	DDDDDD	IIIIII
EE	DD DD	II
EE	DD DD	II
EEEEEEE	DD DD	II
EEEEEEE	DD DD	II
EE	DD DD	II
EE	DD DD	II
EEEEEEEEEE	DDDDDD	IIIIII
EEEEEEEEEE	DDDDDD	IIIIII

EDI - DISK EDITOR

VERSION 3.5

Copyright 1982 by M. Winter
Stuttgart

BENUTZUNGSANLEITUNG
=====

Copyright 1985 by Firma PROFAST
Buchbergstr. 37
Tel.: 07702 - 3246
7712 Blumberg

INHALTSVERZEICHNIS

	Seite
1. EINFÜHRUNG.....	4
1.1 WAS IST EDI.....	4
1.2 SOFT- UND HARDWARE-VORAUSSETZUNGEN.....	4
1.3 STARTEN.....	4
1.4 FEHLER, ANREGUNGEN, KRITIK.....	5
2. DAS KONZEPT.....	6
2.1 DER EDITFILE.....	6
2.2 NOTATION.....	6
2.3 ZEILENNUMMERN.....	7
2.4 SUCHSTRINGS.....	8
2.5 KOMMATA.....	9
2.6 TASTATUR UND STEUERZEICHEN.....	9
3. DIE KOMMANDOS.....	11
3.1 BYE.....	11
3.2 INSERT.....	11
3.3 LINENO-KOMMANDO.....	12
3.4 LIST.....	12
3.5 PRINT.....	13
3.6 DELETE.....	13
3.7 ALTER.....	14
3.8 SAVE.....	15
3.9 EDIT.....	15
3.10 STRING ERSETZEN.....	16
3.11 COPY.....	17
3.12 MOVE.....	17
3.13 NUMBER.....	17
3.14 TABULATOR.....	17
3.15 SEQUENZ-AUFRUF.....	18
3.16 FIND.....	18
3.17 WHEN.....	18
3.18 VARIABLE.....	19
3.19 EXECUTE.....	19
3.20 JUSTIFY.....	19

4.	SYSTEMANPASSUNG.....	21
4.1	LOWER CASE.....	21
4.2	NEWDOS80.....	21
4.3	DELETE DEFAULT VETO.....	21
4.4	LOW UP INVERSION.....	21
4.5	MAKE TAB TO SPACE.....	22
4.6	SUPPRESS SEQ ECHO.....	22
4.7	CHECK PRINTER READY.....	22
5.	AUSDRÜCKE FÜR ZEILEN- UND SPALTENANGABEN....	23
5.1	DIE ITEMS EINES AUSDRUCKS.....	23
5.2	AUSDRÜCKE.....	23
6.	KOMMANDOSEQUENZEN.....	24
6.1	DER SEQUENZ-EDITFILE.....	24
6.2	AUFRUF EINER SEQUENZ.....	24
6.3	FORM EINER KOMMANDOSEQUENZ.....	24
6.4	AUFRUFPARAMETER VON EDI.....	25
6.5	PARAMETER.....	25
6.6	ABLEGEN VON SEQUENZEN AUF DISKETTE.....	26
6.7	BEISPIELE.....	26
6.8	BESCHRÄNKUNGEN.....	27
7.	MITGELIEFERT E KOMMANDOSEQUENZEN.....	28
7.1	FORTRAN/EDI.....	28
7.2	MACRO/EDI.....	28
ANHANG		
A	ZUSAMMENFASSUNG DER KOMMANDOS.....	30
B	ZUSAMMENFASSUNG DER NOTATION.....	31
C	FEHLERMELDUNGEN.....	32

1. EINFÜHRUNG

1.1 WAS IST EDI ?

EDI ist ein extrem schneller und vielseitiger Editor. Das Hauptanwendungsgebiet dürfte die Erstellung von Programmen, Tabellen oder Eingabefiles sein, in gewissem Umfang ist auch Textverarbeitung möglich. EDI behält das gesamte bearbeitete Programm im RAM und benötigt keine Disketten-Zugriffe während des Editierens, was ein wichtiger Grund für die hohe Verarbeitungsgeschwindigkeit von EDI ist. EDI ist sehr komfortabel und bietet viele Möglichkeiten. Trotzdem benötigt EDI nur 4.6 kBytes Speicher, sodaß noch genügend Platz für den zu bearbeitenden Text, den sog. Editfile zur Verfügung steht. EDI ist zeilenorientiert, d.h. ein Befehl spricht immer eine Zeile oder eine Reihe von Zeilen an. Dies bietet für die Programmerstellung mit EDI Vorteile gegenüber zeichenorientierten Editoren, die mehr für Textverarbeitung geeignet sind. Trotzdem gibt es in EDI Möglichkeiten zur Bildschirm - Editierung innerhalb einer Zeile, die sehr effizient und dennoch einfach zu erlernen sind.

1.2 SOFT- UND HARDWARE-VORAUSSETZUNGEN

EDI läuft auf dem TRS-80 Modell 1 unter allen TRSDOS-, NEWDOS23- und NEWDOS80- Betriebssystemen ohne Änderungen. Notwendig ist mindestens ein Diskettenlaufwerk. EDI berücksichtigt den an der Adresse 4049H gespeicherten Wert für die höchste freie Speicheradresse (HIMEM). EDI kommt mit einem auf 16kBytes ausgebauten Speicherbereich aus. Wenn eine Modifikation für Kleinbuchstaben eingebaut ist, dann kann EDI auch Kleinbuchstaben verarbeiten. Eine solche Modifikation ist jedoch nicht Voraussetzung für die Benutzung von EDI.

1.3 STARTEN

Starten Sie EDI einfach durch das DOS - Kommando
EDI[,<seqn>][,<cmdseq>]

Die beiden Parameter können Sie für den Anfang einfach weglassen. Beide Parameter werden in Kapitel 6 näher erläutert. EDI meldet sich sodann mit einer Titelzeile und dem Copyright - Vermerk. Jetzt führt EDI einen kurzen Speichertest durch. Findet EDI unterhalb von HIMEM einen Fehler, so wird ein MEMORY ERROR BELOW HIMEM gemeldet. Findet EDI einen Editfile aus dem vorhergehenden Aufruf, so meldet er EDIT FILE EXISTS. Erklärungen dazu finden Sie in Kapitel 2.1.

1.4 FEHLER, ANREGUNGEN, KRITIK

EDI ist gut durchgetestet. Dennoch kann es sein, daß ein Fehler übersehen worden ist. Falls Sie in EDI irgendwelche Fehler entdecken oder Vorschläge für zukünftige Versionen von EDI haben oder falls Sie Dinge an EDI auszusetzen oder zu verbessern haben, so schreiben Sie bitte direkt an den Programmautor:

Martin Winter
Im Steinengarten 23
7000 Stuttgart 80

2. DAS KONZEPT

2.1 DER EDITFILE

EDI verwaltet das zu bearbeitende Programm oder den Text im RAM. Der dazu verwendete Speicherbereich wird Editfile genannt. Jede Zeile wird im Editfile durch zwei Bytes ergänzt. Vor der Zeile steht die Länge der Text- oder Programmzeile und danach ein Zeilen - Begrenzungszeichen, ein Byte mit dem Wert 0. Diese interne Darstellung ist jedoch für Sie als Benutzer nicht von Bedeutung. Zeilennummern werden nicht abgespeichert. EDI stellt eine Zeilennummer durch Abzählen fest. Dadurch, daß für jede Zeile eine Längenangabe existiert, muß EDI nicht jede Zeile nach dem Begrenzungszeichen durchsuchen, sondern kann direkt zur nächsten springen, sodaß das Abzählen sehr schnell geht. Für 1000 Zeilen benötigt EDI rund 25 Millisekunden. Durch dieses Konzept ist auch sichergestellt, daß die Zeilennummern lückenlos von 1 aufsteigend den Zeilen zugeordnet sind. EDI benötigt deshalb auch kein RENUMBER - Kommando. Die maximale Länge einer Zeile wird von EDI auf 137 Zeichen begrenzt. Längere Zeilen lassen sich auf kaum einem Drucker ausgeben und sind deshalb wenig sinnvoll.

Wenn Sie aus EDI ins Betriebssystem zurückgehen, so versucht EDI, den Editfile in den Speicherbereich über 7000H zu kopieren. Sie können dann einige DOS-Kommandos ausführen, z.B. DIR oder KILL. Wenn Sie dann EDI wieder aufrufen, wird zuerst versucht, den Editfile wieder herzustellen. Dies gelingt nur dann nicht, wenn das Kommando, das Sie inzwischen ausgeführt haben, einen Speicherplatz über 7000H verändert hat. Wenn dies aber nicht der Fall war, so meldet EDI:

EDIT FILE EXISTS

und Sie können nun dort weiterarbeiten, wo Sie vorher aufgehört haben. Der sog. Sequenz - Editfile (s. Kapitel 6) und die Setzung der Tabulatoren sind aber auf jeden Fall verloren. Sie werden nicht auf diese Weise gesichert.

2.2 NOTATION

Zur formalen Darstellung der Befehlsformate von EDI wird in dieser Benutzungsanleitung eine besondere Notation verwendet. Nachfolgend sind die dafür geltenden Regeln aufgeführt:

- Teile in rechteckigen Klammern können wahlweise verwendet oder weggelassen werden.

- Großbuchstaben müssen wie angegeben eingesetzt werden. Sie können, wenn eine Kleinbuchstaben - Modifikation eingebaut und benutzt wird, auch als Kleinbuchstaben eingesetzt werden.
- Teile in Kleinbuchstaben, eingeschlossen in spitzen Klammern, werden vom Benutzer eingesetzt. Sie werden meist im Manual an entsprechender Stelle erläutert. Einige Wichtige seien als Beispiele schon hier genannt:
 <filename> Irgend ein erlaubter Filename.
 <zahl> Eine Dezimalzahl von 0 bis 32767.
- Über die Anwendung von Kommata siehe Kapitel 2.5. Im Zweifelsfalle immer ein Komma einsetzen.
- Von Teilen, die durch senkrechte Striche voneinander getrennt sind, muß genau einer ausgewählt und eingesetzt werden.
- Runde Klammern dienen zum Kennzeichnen von Prioritäten. Sie werden im Kommando immer weggelassen.
- Folgen einem Teil drei Punkte, so kann dieser so oft wie gewünscht oder erforderlich wiederholt werden.
- Ein Kommando darf keine Leerstellen enthalten, außer innerhalb eines Strings.

2.3 ZEILENNUMMERN

Viele Kommandos arbeiten mit Zeilennummern. Eine Zeilennummer kann auf viele verschiedene Arten angegeben werden. Folgendes ist möglich:

- Eine Zahl von 1 bis 32767.
- Der Buchstabe 'F' für 'First' spricht die Zeile 1 an.
- Der Buchstabe 'L' für 'Last' spricht die letzte Zeile an.
- Das Zeichen '*' spricht die aktuelle Zeile an.
- Eine Variable der Form \$n (n = 1..9) spricht diejenige Zeile an, deren Nummer in der angegebenen Variablen enthalten ist.
- Ein arithmetischer Ausdruck, der die 5 oben genannten Möglichkeiten durch + und - beliebig verknüpfen darf.

Beispiele:

- L-4 4 Zeilen vor der letzten, also die fünftletzte Zeile.
 *+2 Die übernächste Zeile nach der aktuellen.

In der oben besprochenen formalen Notation heißt das:

<ziffer19> = 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9
 <item> = <zahl> / F / L / * / \$<ziffer19>
 <ausdruck> = [+ / -] <item> [(+ / -) <item>] ...

Wenn vor dem ersten Item kein Vorzeichen steht, so wird ein Pluszeichen angenommen.

2.4 SUCHSTRINGS

Viele Kommandos erlauben die Verwendung von Suchstrings, um Zeilen näher zu spezifizieren. Ein solcher String hat folgenden Aufbau:

- Er beginnt und endet mit je einem Schrägstrich.
- Wenn er selbst einen Schrägstrich enthalten soll, so muß dieser durch zwei unmittelbar aufeinanderfolgende Schrägstriche dargestellt werden.
- Wenn vor dem Suchstring ein Minuszeichen steht, so wird nicht geprüft, ob die Zeile den nachfolgenden String enthält, sondern, ob sie ihn nicht enthält.

Beispiele:

- /ALPHA/ Die Zeile muß das Wort ALPHA enthalten.
 /17//4/ Die Zeile muß die Zeichen 17/4 enthalten.
 Beachten Sie, daß im Suchstring zwei Schrägstriche stehen müssen.
 -/ / Die Zeile darf keine Leerstelle enthalten.
 -// Die Zeile darf überhaupt kein Zeichen enthalten.

Ein Suchstring kann durch eine Spaltenangabe ergänzt werden. Dies können ein oder zwei Zahlen oder arithmetische Ausdrücke sein. Wird nur ein Ausdruck angegeben, so heißt das, daß der angegebene String in dieser Spalte beginnen muß. Durch die Angabe von zwei Ausdrücken kann man einen Spaltenbereich festlegen, der das erste Zeichen des Strings enthalten muß. Für arithmetische Ausdrücke in einer solchen Spaltenangabe gilt einschränkend, daß sie nicht die Zeichen F, L oder * enthalten dürfen und daß ihr Wert kleiner als die maximale Zeilenlänge, also kleiner als 137 sein muß. Beispiele sind:

- /BETA/,5 Sucht BETA mit B in Spalte 5.

/1234/,1,7 Sucht 1234 mit 1 in Spalte 1 bis 7.
//,45 Die Spalte 45 muß irgendein Zeichen enthalten.

Beachten Sie in den obigen Beispielen, daß Sie mit dem Leerstring // abprüfen können, ob irgendein Zeichen in der angegebenen Position steht. Der Leerstring stimmt also mit jedem Zeichen, auch dem Leerzeichen überein. Er wird nur dann als nicht gefunden betrachtet, wenn die Zeile vor der angegebenen Spalte zuende ist. Das letzte Beispiel wählt also Zeilen aus, die mindestens 45 Zeichen enthalten.

Hinter diesen Angaben kann noch der Buchstabe U folgen. Dieser sog. Unit-Parameter gibt an, daß der Suchstring im Text von nicht-alphanumerischen Zeichen eingeschlossen sein muß. Dies ist z.B. sinnvoll, wenn Sie in einem Programm nach Variablenamen suchen wollen. Auch der Beginn und das Ende einer Zeile gelten als nicht-alphanumerische Zeichen.

2.5 KOMMATA

Kommata dienen dazu, die einzelnen Parameter eines Kommandos voneinander zu trennen. Der Einfachheit halber können diese Kommata aber überall entfallen, wo sie zur Trennung nicht unbedingt erforderlich sind. Notwendig sind sie nur:

- Zwischen zwei Zahlen, z.B. zwei Zeilennummern
- Nach einem Filenamen

An allen anderen Stellen sind sie optional und können weggelassen werden. Der Übersichtlichkeit halber sind in diesem Manual Kommata immer mit angegeben.

2.6 TASTATUR UND STEUERZEICHEN

EDI benutzt folgende Tasten abweichend vom üblichen Gebrauch:

- | | |
|------|---|
| Sh ↑ | Darstellung des Pfeiles nach oben oder öffnende eckige Klammer. |
| Sh ↓ | Darstellung des Pfeiles nach unten oder sog. Backslash. |
| Sh ← | Darstellung des Pfeiles nach links oder schließende eckige Klammer. |
| Sh → | Darstellung des Pfeiles nach rechts oder Zirkumflex. Mit diesen vier Tasten können also die Pfeile als Zeichen eingegeben werden. |

- Normalerweise funktioniert diese Taste wie üblich, d.h. als Tabulator zu jeder achten Bildschirmposition. Im Insert-Kommando kann man die Wirkung dieser Taste durch das I-Kommando beeinflussen.
- ← Löschen des letzten eingegebenen Zeichens.
- ↓ Zeilenvorschub. Wird zwar auf dem Bildschirm durchgeführt, aber nicht in die Eingabe übernommen.
- Clear Gesamte Eingabezeile löschen (normalerweise durch Sh ←) oder Kommando abbrechen.
- Sh Clear Bildschirm löschen (normalerweise Clear).
- Shift Ausgabe anhalten, solange Taste gedrückt.

Im Alter-Modus gelten folgende Definitionen:

- ← Cursorbewegung nach links.
- Cursorbewegung nach rechts.
- ↓ Löschen von Zeichen.
- ↑ Ein- und Ausschalten des Insert-Modus.
- Sh Clear Cursor zum Anfang bzw. Ende der Zeile bewegen.

In den obenstehenden Erläuterungen bedeutet Sh, daß zu der angegebenen Taste die Shift-Taste gedrückt werden muß. Alle nicht erwähnten Tasten haben die normalen Bedeutungen. Um die angegebenen Umdefinitionen zu erreichen, schaltet sich EDI in die Tastatureingabe - Routine ein. Damit dies wieder rückgängig gemacht wird, muß EDI über den Bye-Befehl verlassen werden. Wenn man EDI über Debug mit G402D oder über MINI-DOS mit MDOBORT verläßt, sind diese Umdefinitionen noch vorhanden und zwar solange, wie der entsprechende Speicherbereich nicht verändert wird. Wird dies aber getan, so ist im günstigsten Fall ein Neu-Booten des Systems die Folge. Deshalb wird davon dringend abgeraten.

3. DIE KOMMANDOS

=====

3.1 BYE

Dieses Kommando benötigen Sie, um sich von EDI zu verabschieden. Das Format ist einfach ein

B

Es dürfen keine weiteren Parameter angegeben sein, sonst meldet EDI einen Fehler. EDI versucht nun, den Editfile in den Speicherbereich ab 7000H zu verschieben. Wenn dazu genügend Platz vorhanden ist, geht EDI danach ins DOS. Wenn Sie EDI nun erneut aufrufen, können Sie an dem alten Editfile weiterarbeiten. War aber nicht genügend Platz frei, um den Editfile entsprechend zu verschieben, so fragt EDI:

CANT MOVE EDITFILE, QUIT ANYWAY (Y/N) ?

Tippen Sie jetzt ein N für No, dann meldet sich EDI wieder wie gewohnt mit den beiden Punkten. Geben Sie aber Y für Yes ein, dann können Sie nicht mehr damit rechnen, beim erneuten Aufruf von EDI den alten Editfile noch behalten zu haben.

3.2 INSERT

Mit diesem Kommando werden Sie im allgemeinen Zeilen eingeben. Das Format sieht zwei Möglichkeiten vor. Wenn Sie neue Zeilen an das Ende des bisherigen Editfiles anfügen wollen oder wenn noch überhaupt kein Text im Editfile ist, dann tippen Sie einfach ein

I

Wenn Sie mitten in den Editfile einfügen wollen, so tippen Sie

I,<ausdruck>

wobei Sie die genaue Erklärung für <ausdruck> in Kapitel 5 finden und eine kurze Erläuterung in Kapitel 2.3. Hier sei nur soviel gesagt, daß Sie für <ausdruck> zum Beispiel eine Zeilennummer einsetzen können und daß Sie mit dem I-Kommando dann vor dieser Zeile einfügen können. In beiden Fällen meldet EDI dann eine 5-stellige Zeilennummer. Sie können jetzt eine Zeile eingeben. Schließen Sie die Zeile mit <enter> ab. Es erscheint sofort die nächste Zeilennummer und so fort. Wollen Sie keine weiteren Zeilen mehr einfügen, so geben Sie eine leere Zeile ein, d.h. drücken Sie zweimal direkt hintereinander auf die <enter>-Taste.

Nach Beendigung des I-Kommandos ist die aktuelle Zeile die erste eingefügte Zeile. Sie können dann durch ein einfaches L-Kommando ohne Parameter (s. 3.4) die gerade eingegebenen Zeilen auflisten lassen. Nocheinmal sei darauf hingewiesen, daß Insert vor der angegebenen Zeilennummer einfügt, wenn <ausdruck> spezifiziert ist.

Die Pfeiltaste nach rechts dient im I-Kommando als Tabulator. Die Tabulatorpositionen können mit dem T-Kommando gesetzt und abgefragt werden. Beachten Sie hierzu Kapitel 3.14.

3.3 LINENO-KOMMANDO

Mit diesem Kommando können Sie eine einzelne Zeile neu eingeben. Es lautet einfach:

`<ausdruck>=<text>`

Lesen Sie das Kapitel 2.3 für einfache Beispiele dafür, was Sie für `<ausdruck>` einsetzen können. Häufig wird man hier einfach die Nummer der einzugebenden Zeile einsetzen. `<text>` ist eine beliebige Folge von Buchstaben, Ziffern und Sonderzeichen. Existiert die angegebene Zeilennummer noch nicht, so wird die Zeile hinten an den Editfile angehängt, sonst wird die alte Zeile durch den neuen Inhalt ersetzt.

3.4 LIST

Dieses Kommando dient zum Auflisten von Zeilen auf dem Bildschirm. Sie werden es wohl am häufigsten benötigen. Das Format ist:

`L[,<bereich>][,<suchstr>]`

Beide Angaben sind optional und können deshalb weglassen werden. `<bereich>` ist folgendermaßen definiert:

`<bereich> = A / (<ausdruck> [, <ausdruck>])`

d.h. entweder ein A für All, es wird dann der gesamte Editfile angesprochen oder ein `<ausdruck>`, dann ist nur die dadurch bezeichnete Zeile gemeint, oder zwei Ausdrücke, dann sind alle Zeilen von der ersten bis zur zweiten Angabe einschließlich gemeint. Ist `<bereich>` überhaupt nicht angegeben, dann wird der Bereich von der aktuellen Zeile bis 14 Zeilen weiter angenommen, d.h. L entspricht L,*,*+14. Dadurch läßt sich bequem jeweils eine Bildschirmseite listen. `<suchstr>` wählt, wenn angegeben, aus dem spezifizierten Zeilenbereich diejenigen Zeilen aus, die auch noch dieser Angabe entsprechen. Dadurch beinhaltet das List - Kommando in EDI auch gleich eine Find - Möglichkeit, die bei anderen Editoren oft ein eigener Befehl ist.

Beispiele:

L,A,/HUGO/ Listet alle Zeilen, die das Wort HUGO enthalten.

L,*,L Listet von der aktuellen Zeile bis zum Schluß.

L,A,/C/,1,U Listet alle Zeilen mit einem C in Spalte 1 und einem Sonderzeichen oder einem Leerzeichen in Spalte 2 (Unit - Parameter).

Wird die Ausgabe zu lang und läuft zu schnell durch, so haben Sie zwei Möglichkeiten. Sie können die Ausgabe anhalten, indem Sie die Shift - Taste drücken und niedergedrückt halten. Wenn Sie diese Taste wieder loslassen, wird die Ausgabe fortgesetzt. Zum anderen können Sie aber auch die Clear - Taste drücken. Mit der Clear - Taste kann man jedes Kommando abbrechen, bei dem dies sinnvoll ist, so auch das List - Kommando.

3.5 PRINT

Dieses Kommando hat genau die gleichen Parameter und Eigenschaften wie LIST. Bei PRINT geht die Ausgabe jedoch auf den Drucker. Ist kein Drucker angeschlossen oder kein Papier eingelegt, dann meldet EDI:

PRINTER NOT READY

und bricht das P-Kommando ab. Diese Meldung kann durch setzen eines Parameters im System Configuration Mode (s. 4.7) erlaubt oder unterdrückt werden. Der Vollständigkeit halber hier noch einmal das Format:

PC,<bereich>JC,<suchstr>

3.6 DELETE

Das Delete-Kommando dient zum Löschen von Zeilen aus dem Editfile. Das Format ist:

DC,<bereich>JC,<suchstr>JC,V

Die Angaben <bereich> und <suchstr> sind schon aus Kapitel 3.4 bekannt. Der Default bei fehlender Bereichsangabe ist hier die aktuelle Zeile. Das Kommando D ohne Parameter löscht also nur diese Zeile. Neu ist hier die Möglichkeit, ein V als letzten Parameter anzugeben. Dieser sogenannte Veto-Parameter bewirkt, daß EDI die angegebenen Zeilen zuerst anzeigt und den Benutzer fragt, ob sie tatsächlich gelöscht werden sollen. EDI zeigt dabei ein Fragezeichen unter der zu löschenden Textzeile an. Mögliche Antworten auf dieses Fragezeichen sind:

Y Ja, diese Zeile soll gelöscht werden.

N Nein, diese Zeile soll nicht gelöscht werden.

C Continue, alle weiteren Zeilen sollen gelöscht werden, ohne jedesmal die Zeile und das Fragezeichen auszugeben. Dadurch wird der Veto-Parameter außer Kraft gesetzt.

S Stop, keine weiteren Zeilen sollen gelöscht werden.

D Display, alle weiteren Zeilen, die in dem betreffenden Kommando angesprochen werden, werden auf dem Bildschirm angezeigt und dann ohne Frage an der Benutzer gelöscht. Auch die Ausgabe bei Display kann durch die Shift - Taste vorübergehend angehalten werden.

Die Benutzung des Veto-Parameters wird dringend angeraten. Um ihn nicht jedesmal angeben zu müssen, kann man EDI dazu veranlassen, bei Delete automatisch Veto anzunehmen. Lesen Sie dazu Kapitel 4.3. Auch das Delete - Kommando können Sie durch Clear abbrechen. Im Veto - Modus unterscheiden sich Clear und S dadurch, daß S für Stop nur das Delete - Kommando beendet, während Clear auch eine Kommando-sequenz abbricht, in der das D-Kommando möglicherweise steht.

3.7 ALTER

Mit diesem Kommando können Sie Änderungen an einzelnen Zeilen durchführen. Es bietet einfache, aber komfortable Bildschirm - Editiermöglichkeiten. Das Format ist:
AC, <bereich>J, <suchstr>

Die Beschreibung der Parameter <bereich> und <suchstr> entnehmen Sie bitte wieder dem Kapitel 3.4. Der Default bei fehlender Bereichsangabe ist hier die aktuelle Zeile. Jede durch die Angaben spezifizierte Zeile wird nun auf dem unteren Viertel des Bildschirms angezeigt, dazu ein Cursor in Form eines blinkenden Rechteckes, das mit dem Zeichen auf dieser Position abwechselt. Sie können nun diesen Cursor durch die Tasten Pfeil rechts und Pfeil links in der Zeile verschieben. Diese Tasten haben Dauerfunktion und werden bei längerem Niederdrücken in schneller Folge wiederholt. Wird eine Zeichentaste gedrückt, so können dadurch fehlerhafte Stellen in der Zeile überschrieben werden. Drücken Sie den Pfeil nach unten, so wird das Zeichen, auf dem der Cursor gerade steht, gelöscht und die übrigen Zeichen rücken von rechts nach. Auch diese Taste hat Dauerfunktion. Drücken Sie den Pfeil nach oben, so ändert sich der Cursor zu einem kleineren Quadrat unterhalb der Zeile. Dies ist der Insert - Modus. Jedes Zeichen, das jetzt eingegeben wird, wird an der jeweiligen Cursorposition eingeschoben. Alle Zeichen rechts vom Cursor rücken dabei nach rechts. Erneutes Drücken des Pfeiles nach oben schaltet den Insert - Modus wieder ab. Drücken Sie Shift Clear, so geschieht folgendes: Befindet sich der Cursor auf dem ersten Zeichen der Zeile, so wird er hinter das letzte Zeichen bewegt, sonst wird er auf das erste Zeichen der Zeile gesetzt. Damit kann man sehr schnell den Cursor über die ganze Zeile hinweg bewegen.

Man kann den Cursor bis maximal ein Zeichen hinter das Zeilenende bewegen. Wenn er hier steht, kann man Zeichen an die Zeile anfügen, ohne in den Insert - Modus gehen zu müssen. Die Zeile kann nie länger als 137 Zeichen werden.

Wenn Sie alle Änderungen an der jeweiligen Zeile angebracht haben, drücken Sie entweder <enter>, um die Änderungen in den Editfile zu übernehmen und die nächste Zeile anzuzeigen oder Sie betätigen die Clear - Taste, um das Alter - Kommando abzubrechen. Dabei wird auch die Änderung an der gerade bearbeiteten Zeile ignoriert. Üben Sie den Umgang mit Alter, dieses Kommando ist ein wichtiges Hilfsmittel in EDI.

3.8 SAVE

Mit diesem Kommando können Sie eingegebene oder geänderte Programme auf Diskette schreiben. Das Format ist:
S[<co>],<filename>[,<bereich>][, <suchstr>]

<co> kann entweder ein Pluszeichen sein, dann werden die EDI - Zeilennummern lesbar, d.h. mit Bit 7 = 0 mit auf dem File abgespeichert. Eine andere Möglichkeit ist die Angabe eines Minuszeichens. Dadurch wird die Ausgabe von Zeilennummern vollständig unterdrückt, z.B. für Texte. Läßt man diesen Parameter ganz weg, so speichert EDI Zeilennummern mit ab, jedoch mit gesetztem 7. Bit. Solche Zeilennummern akzeptieren die meisten Compiler und Assembler, so z.B. Microsoft's M80 - Assembler und F80 - FORTRAN - Compiler. Der Default für den Bereich ist All, sodaß im allgemeinen das Kommando S,<filename> genügt.

3.9 EDIT

Mit Edit können Sie Files von Diskette lesen. Das Format ist:
E,<filename>[,<ausdruck>]

Wenn der Editfile noch keine Zeilen enthält, dann ist <ausdruck> bedeutungslos. Sind jedoch schon Zeilen im Editfile enthalten, so kann man durch <ausdruck> angeben, vor welcher Zeile der File eingefügt werden soll. Default ist Einfügen hinter der letzten Zeile.

Das Edit - Kommando überliest alle Steuerzeichen auf dem File, außer hex 00, dem Zeilenbegrenzer. Durch eine Option kann man außerdem veranlassen, daß Tabulatorzeichen (hex 09) durch Leerstellen ersetzt werden. Siehe dazu Kapitel 4.5. Enthält der File Zeilen mit mehr als 137 Zeichen, so bricht EDI den Einlesevorgang mit der Meldung ILLEGAL TEXT FILE ab.

3.10 STRING ERSETZEN

Dies ist das fähigste und leider auch komplizierteste Kommando in EDI. Damit ist es möglich, im Text bestimmte Wörter zu finden und durch andere zu ersetzen. Das allgemeine Format ist:

```
<suchstr1>=<string>[,<bereich>][,<suchstr2>][C,V]
```

Dieses Kommando führt folgende Funktionen aus: Es wählt aus den Zeilen, die in <bereich> angegeben sind, diejenigen aus, die auch den <suchstr2> enthalten, sofern dieser angegeben ist. Diese Zeilen werden dann daraufhin untersucht, ob sie auch den <suchstr1> enthalten. Wenn dies der Fall ist, wird jedes Auftreten von <suchstr1> in dieser Zeile durch <string> ersetzt. Der Unterschied zwischen einem <suchstr> und <string> ist der, daß ein <suchstr> noch zusätzliche Spaltenangaben enthalten darf, während ein <string> ohne solche Spaltenangaben sein muß. Wird der Veto - Parameter gesetzt, so hat er eine ähnliche Wirkung wie bei Delete. Lesen Sie dazu bitte Kapitel 3.6. Die folgenden Beispiele erklären auch einige Sonderfälle:

/EINS/=ZWEI/,A	Ersetze im ganzen Editfile das Wort EINS durch ZWEI. Beachten Sie aber, daß dabei auch das Wort EINSTEIN zu ZWEITEIN werden kann! Wollen Sie dies verhindern, so geben Sie den Unit-Parameter an:
/EINS/,U=ZWEI/,A	Hier muß das Wort EINS als Unit auftreten, damit es ersetzt wird.
/C/,1=//,*,L	Von der aktuellen Zeile bis zum Schluß entferne jedes C in Spalte 1.
//,1,5=//,A,/DREI/	Entferne in allen Zeilen, die das Wort DREI enthalten, die ersten 5 Spalten, egal, was sie enthalten. Diese Sonderfunktion wird ausgeführt, wenn beide Strings rechts und links vom Gleichheitszeichen Leerstrings sind.
//=/ /,34	Erzeugt in Zeile 34 einen gesperrten Text, d.h. zwischen je zwei Zeichen wird ein Leerzeichen eingefügt. Beachten Sie, daß der erste String ein Leerstring ist und deshalb an jeder Position gefunden wird, an der irgend ein Zeichen steht.

Üben Sie den Umgang mit dem String - Ersetze - Befehl. Sie können damit sehr effizient arbeiten.

3.11 COPY

Dieser Befehl kann Zeilen innerhalb des Editfiles kopieren. Sein Format ist:

C, <ausdruck>C, <bereich>]C, <suchstr>]C, V]

<ausdruck> muß angegeben werden, alle anderen Angaben sind optional. Der Default für <bereich> ist die aktuelle Zeile.

Copy kopiert die durch <bereich> und <suchstr> spezifizierten Zeilen vor die Zeile, deren Nummer <ausdruck> angibt. Die Funktion des Veto - Parameters wurde schon in Kapitel 3.6 besprochen. Copy entfernt die kopierten Zeilen nicht an ihrem ursprünglichen Ort. Nach der Ausführung des Copy - Kommandos steht der Zeiger für die aktuelle Zeile auf der letzten kopierten Zeile.

3.12 MOVE

Dieser Befehl hat das gleiche Format und eine ganz ähnliche Funktion wie Copy:

M, <ausdruck>C, <bereich>]C, <suchstr>]C, V]

Move löscht die kopierten Zeilen jedoch an ihrem alten Platz, sodaß sich die Gesamtgröße des Editfiles nicht verändert.

3.13 NUMBER

Dies ist ein Hilfskommando zur Bestimmung von Spaltennummern. Sein Format ist einfach ein

N

Es zeigt eine Zeile mit Ziffern an, anhand derer leicht die Spaltenposition eines Zeichens festgestellt werden kann.

3.14 TABULATOR

Mit dem T-Kommando können Sie Tabulatorstops für die Eingabe im I-Kommando setzen und abfragen. Der Tabulator ist nur innerhalb des I-Kommandos in Kraft. Zur Abfrage der gesetzten Tabulatoren geben Sie das Kommando

T

ein. Es werden neun Zahlen angezeigt, die den neun möglichen Tabulatoren entsprechen. Die Standard - Setzung ist:

7 1 1 1 1 1 1 1

Diese Einstellung ist besonders für FORTRAN - Programme sinnvoll. Beachten Sie, daß Tabulatorstops, die auf eine kleinere Spalte als einer ihrer Vorgänger zielen, nicht beachtet werden, sodaß in der oben genannten Einstellung nur ein Tabulator zur siebten Spalte wirksam ist. Das Setzen des Tabulators geschieht durch
T,<spalte> ...

Beispielsweise wäre für PASCAL - Programme folgende Einstellung nützlich:

T,4,7,10,13,16,19,22,25,28

Damit läßt sich ein gut strukturiertes Programm erstellen. Wenn Sie weniger als neun Tabulatoren angeben, so werden die übrigen nicht verändert. Geben Sie mehr als neun Positionen ein, dann werden die überflüssigen Angaben einfach ignoriert. Die Positionsangaben können auch einfache Ausdrücke sein, wie sie in Kapitel 5 für Spaltenangaben erläutert sind.

Der Tabulator kann nicht über das Zeilenende hinweg arbeiten. Die größte angebbare Position ist deshalb die 58. Spalte. Sechs Spalten sind ja schon durch die Zeilennummer belegt und stehen deshalb bei Insert auf der ersten Bildschirmzeile nicht mehr zur Eingabe zur Verfügung.

3.15 SEQUENZ - AUFRUF

Mit diesem Kommando kann man Kommandosequenzen ausführen. Lesen Sie bitte zur weiteren Erklärung das Kapitel 6.

3.16 FIND

Find benötigt man normalerweise nur innerhalb von Kommandosequenzen. Das Format ist

FC,<bereich>][,<suchstr>]

Find setzt den Zeiger für die aktuelle Zeile auf die erste Zeile, die den angegebenen Spezifikationen entspricht. Default für <bereich> ist All. Entspricht überhaupt keine Zeile den Angaben, so stellt F diesen Zeiger auf -1, einen unzulässigen Wert für eine Zeilennummer. Mit dem When - Kommando (s. 3.17) kann diese Bedingung abgefragt werden.

3.17 WHEN

Dieses Kommando wird im allgemeinen ebenfalls nur in Kommandosequenzen benötigt. Es gestattet es, ein Kommando nur unter bestimmten Bedingungen auszuführen. Sein Format ist:

W,<bed>,<ausdruck>,<edicmd>

Dabei wird geprüft, ob <ausdruck> der Bedingung <bed> genügt. Nur wenn dies der Fall ist, wird das EDI - Kommando <edicmd> ausgeführt. <edicmd> kann auch ein weiteres When - Kommando sein. <bed> kann sein:

- + Teste, ob <ausdruck> positiv oder Null ist.
- Teste, ob <ausdruck> negativ ist.
- Z Teste, ob <ausdruck> Null ist.
- Z Teste, ob <ausdruck> ungleich Null ist.

3.18 VARIABLE

EDI bietet die Möglichkeit, Zeilen- und Spaltennummern in neun Variablen abzulegen. Diese Variablen können mit dem V-Kommando gesetzt und abgefragt werden. Das Format kann sein:

V<ziffer19>

Dabei ist <ziffer19> eine Ziffer von 1 bis 9. EDI zeigt dann den momentanen Wert dieser Variablen an. Oder:

V<ziffer19>=<ausdruck>

EDI weist der Variablen dann den Wert des Ausdrucks zu und zeigt den neuen Wert an (s. auch 4.6).

3.19 EXECUTE

Dieses Kommando funktioniert nur unter NEWDOS80! Dann kann man damit auf einfache Weise MINI-DOS - Kommandos ausführen. Das Format ist:

X,<doscmd>

Dabei ist <doscmd> irgend ein DOS - Kommando, das unter MINI-DOS ausgeführt werden kann wie z.B. KILL oder DIR. X,MDBORT und X,MORET sind verboten. Unter anderen Betriebssystemen müssen Sie dazu EDI verlassen, das DOS - Kommando ausführen und EDI erneut starten. Beachten Sie bitte auch das Kapitel 4.2. Wenn der NEWDOS80 - Parameter nicht auf Y steht, erzeugt das X-Kommando einen ILLEGAL COMMAND Fehler.

3.20 JUSTIFY

Mit diesem Kommando kann man Zeilen verlängern oder verkürzen. Das Format ist:

J,<spalte>C,<bereich>J[,<suchstr>]

Der Default für <bereich> ist All, also alle Zeilen des Editfiles, <spalte> muß angegeben sein. Die Wirkung ist wie folgt: Alle angesprochenen Zeilen werden untersucht, ob sie länger oder kürzer als die angegebene Zahl <spalte> sind. Ist eine Zeile kürzer, als hier angegeben, so werden hinten an diese Zeile Leerstellen angehängt, bis sie die angegebene Länge hat. Ist eine Zeile länger, so wird untersucht, ob die Zeile hinten mit Leerstellen endet. Ist das der Fall, so werden diese Leerstellen soweit entfernt, bis die Zeile die angegebene Länge erreicht hat. Befinden sich am Ende keine Leerstellen mehr, so wird die Zeile nicht weiter gekürzt. Dieses Kommando ist nützlich, wenn man ein Assemblerprogramm nachträglich kommentieren will. Man erweitert dann zuerst alle Zeilen auf eine feste Länge, z.B. auf 35 Zeichen, durch das Kommando J,35. Dann fügt man im Alter- Modus die Kommentare an. Dort kann man dann Shift Clear benutzen, um schnell zum Ende einer Zeile zu gelangen. Um danach die Leerstellen an denjenigen Zeilen wieder zu entfernen, die nicht kommentiert worden sind, gibt man das Kommando J,1. Dadurch werden alle Leerstellen am Ende einer Zeile wieder entfernt.

4. SYSTEMANPASSUNG

EDI kann auf einfache Weise an unterschiedliche Systeme und Gegebenheiten angepaßt werden. Hierzu gibt es einen besonderen sogenannten 'System Configuration Mode', den man aufruft, indem man beim Starten von EDI die Shift - Taste niedergedrückt hält. EDI zeigt dann in einer Tabelle sieben Systemparameter an. Diese Parameter können Sie nun ändern. EDI trägt die Parameter in den File EDI/CMD ein. Dazu muß EDI aber auch unter diesem Namen auf Diskette liegen. Wenn EDI diesen File nicht finden kann, meldet er CANT ZAP EDI und ändert die Parameter nicht.

4.1 LOWER CASE

Dieser Parameter muß auf Y gesetzt werden, wenn Sie eine Kleinbuchstaben - Modifikation in Ihrem TRS80 installiert haben und benutzen wollen. Drücken Sie dazu auf die Frage CHANGE PARAMETERS ? die Taste Y. Geben Sie dann die Nummer des Parameters, in diesem Fall eine 1 und den Wert, d.h. entweder Y oder N ein. EDI setzt den entsprechenden Parameter und zeigt erneut die Tabelle an.

4.2 NEWDOS80

Setzen Sie diesen Parameter nur dann auf Y, wenn Sie EDI unter einem NEWDOS80 - Betriebssystem benutzen. Dieser Parameter schaltet insbesondere das X - Kommando (s. 3.18) ein und aus.

4.3 DELETE DEFAULT VETO

Wenn Sie diesen Parameter auf Y setzen, dann wird bei jedem Delete - Kommando automatisch Veto angenommen. Es ist sehr nützlich, hier ein Y einzugeben, um versehentliches Löschen zu vermeiden! Steht hier ein N, so muß der Veto - Parameter explizit angegeben werden, wenn er gewünscht wird. Dieser Parameter ist innerhalb von Kommandosequenzen nicht wirksam.

4.4 LOW UP INVERSION

Manche Betriebssysteme und Kleinbuchstaben - Driver verlangen diesen Parameter, um Großbuchstaben bei gedrückter Shift - Taste darzustellen und nicht umgekehrt. Wenn dieser Parameter auf Y steht, dann werden Groß- und Kleinbuchstaben vertauscht. Setzen Sie diesen Parameter so, wie es Ihren Erfordernissen und Ihrem Betriebssystem entspricht.

4.5 MAKE TAB TO SPACE

Wenn hier ein Y steht, dann werden Tabulatorzeichen (hex 09) in einem Eingabefile vom Edit - Kommando in Leerstellen umgewandelt. Steht hier ein N, dann werden Tabulatorzeichen ignoriert. Wählen Sie den richtigen Wert gemäß Ihren Erfordernissen.

4.6 SUPPRESS SEQ ECHO

Normalerweise, d.h. wenn hier ein N steht, zeigt EDI alle Befehle aus einer Sequenz, die es gerade ausführt, an. Steht hier ein Y, dann wird diese Ausgabe unterdrückt, ebenso wie die Ausgabe des neuen Wertes einer Variablen bei einer Wertzuweisung mit dem V-Kommando. Setzen Sie diesen Parameter nur dann auf Y, wenn Sie voll durchgetestete Kommando - Sequenzen benutzen, da er die Fehlersuche sehr behindern kann.

4.7 CHECK PRINTER READY

Wenn hier Y steht, dann überprüft das P-Kommando, ob ein Drucker angeschlossen ist oder nicht und gibt ggf. eine Fehlermeldung aus. Das kann aber stören, wenn man einen Drucker nicht an der Standard- Schnittstelle angeschlossen hat, sondern z.B. einen Drucker über eine serielle Schnittstelle betreibt. Dann sollte man hier ein N setzen, um die Fehlermeldung zu umgehen. Auch beim Video Genie muß hier N gesetzt werden, da hier der Drucker nicht Memory Mapped angeschlossen ist.

5. AUSDRÜCKE FÜR ZEILEN- UND SPALTENANGABEN

5.1 DIE ITEMS EINES AUSDRUCKS

Ein Ausdruck ist für EDI eine Verbindung von einem oder mehreren Items durch Plus- oder Minuszeichen. Bei Bedarf kann auch vor dem ersten Item ein Vorzeichen stehen. Die Items eines Ausdrucks sind also die Summanden. Folgende Formen sind möglich:

- F Dieses Item hat immer den Wert 1 und bezeichnet damit die erste Zeile des Editfiles. Statt eines F's könnte immer auch eine 1 stehen, jedoch spart man durch die Verwendung von F in manchen Fällen ein Komma ein.
- L Der Wert dieses Items ist die Nummer der letzten Zeile des Editfiles. Ist noch überhaupt keine Zeile im Editfile, so hat L den Wert 0. In diesem Fall erzeugt z.B. das List - Kommando der Form L,L einen ILLEGAL NUMBER Fehler, da L,0 nicht erlaubt ist.
- * Der Wert dieses Items ist die Nummer der aktuellen Zeile. Dieser Wert ist immer legal außer nach einem Find - Kommando, welches keine Zeile gefunden hat. In diesem Fall nimmt * den Wert -1 an.
- \$n wobei n eine Ziffer von 1 bis 9 ist. Dies bezeichnet eine von neun Variablen, die mit dem V-Kommando (s. 3.17) gesetzt und abgefragt werden können. Diese Variablen können beliebige Werte im Bereich von -32768 bis 32767 annehmen.
- nnn stellt eine bis zu 5-stellige Dezimalzahl dar, die ebenfalls als Item verwendet werden kann. nnn muß kleiner als 32768 sein.

5.2 AUSDRÜCKE

Aus diesen fünf Items können jetzt Ausdrücke gebildet werden. Klammern können nicht gesetzt werden. Beispiele sind:

- *-5 Fünf Zeilen vor der aktuellen Zeile.
- L-7 Sieben Zeilen vor der letzten, also die achtletzte.
- \$1*-2 Suche die zweite Zeile vor der aktuellen (*-2) und gehe von da aus so viele Zeilen nach unten, wie in \$1 steht.

Sie können Ausdrücke auch für Spaltenangaben hinter einem Suchstring oder im I-Kommando verwenden. Dabei sind aber die Items F, L und * nicht erlaubt.

6. KOMMANDOSEQUENZEN *****

6.1 DER SEQUENZ - EDITFILE

Neben dem normalen Editfile führt EDI noch einen zweiten, normalerweise 512 Bytes langen Editfile. Dort können Kommandosequenzen abgespeichert werden und bei Bedarf mit dem Q-Kommando aufgerufen werden. Diesen Sequenz - Editfile spricht man an, indem man die normalen EDI - Kommandos benutzt und nur vor die Angabe der Zeilennummer oder des Zeilenbereiches ein @ setzt. Bei Move und Copy kann der @ sowohl vor der Bereichsangabe stehen als auch vor der Zeilennummer, an die hinkopiert werden soll. Beispiele sind:

- L,@,A Listet den gesamten Sequenz - Editfile.
- A,@,2 Ändert die zweite Zeile aus dem Sequenz - Editfile.
- @,3=L,A Eingabe des Kommandos L,A in die dritte Zeile des Sequenz - Editfiles.
- D,@,5,L Löscht den Sequenz - Editfile von der fünften bis zur letzten Zeile.
- C,@,1,5 Kopiert die fünfte Zeile des Sequenz - Editfiles vor die erste Zeile.
- C,1,@,5 Tut genau dasselbe wie das vorhergehende Kommando.

6.2 AUFRUF EINER SEQUENZ

Eine Kommandosequenz ruft man mit dem Q-Kommando auf. Sein Format ist:

Q,<ausdruck>[,<parameter>] ...

<ausdruck> gibt die Nummer der Zeile im Sequenz - Editfile an, die als Kommandosequenz ausgeführt werden soll. Vor <ausdruck> darf kein @ stehen. Die Angabe weiterer Parameter ist optional und wird in Kapitel 6.5 erläutert.

6.3 FORM EINER KOMMANDOSEQUENZ

Eine Kommandosequenz besteht aus einem oder mehreren ganz normalen EDI - Kommandos, die mit Ausrufe - Zeichen voneinander getrennt sind. Zwischen den Kommandos sind keine Leerstellen erlaubt. Beispiel für eine gültige Kommandosequenz:

- A,*!L,*-6,*+6 Geht mit der aktuellen Zeile in den Alter - Mode und listet danach die Umgebung der geänderten Zeile.

Diese Sequenz könnte man durch

- @,1=A,*!L,*-6,*+6

In den Sequenz - Editfile abspeichern und mit

Q,1

aufrufen. Sequenzen dürfen selbst weitere Sequenzen aufrufen. Die Anzahl der Schachtelungen ist beliebig und wird nur durch den Speicherplatz begrenzt, der zum Ablegen von Parametern und Pointern von EDI intern im Sequenz - Editfile benötigt wird. Es ist zu beachten, daß schon vor der Ausführung des letzten Kommandos einer Sequenz EDI um eine Schachtelungstiefe nach oben rückt. Ist das letzte Zeichen einer Kommandosequenz ein Ausrufezeichen, dann springt EDI beim Erreichen des Zeilenendes zur nächsten Zeile des Sequenz - Editfiles und führt diese aus. Gibt es keine nächste Zeile, dann wird das Ausrufezeichen nicht beachtet.

Befindet sich EDI in der Ausführung einer Sequenz, so darf kein Kommando gegeben werden, das den Sequenz - Editfile in irgend einer Weise verändern könnte. Wird dies versucht, so meldet EDI: ILLEGAL IN SEQ-MODE und bricht die Sequenz ab. Grundsätzlich kann man sagen, daß eine Kommandosequenz bei jedem Fehler, außer den nichtfatalen Fehlern wie NO SUCH LINE und LINE TRUNCATED abgebrochen wird. Die Meldung NO SUCH LINE wird im Sequenz - Modus jedoch unterdrückt. Sie wäre dort üblicherweise wenig informativ und würde mehr stören als helfen.

6.4 AUFRUFPARAMETER VON EDI

In Kapitel 1.3 wurde angedeutet, daß beim Starten von EDI zwei Parameter angegeben werden können. Die Form des Aufrufs ist:

EDI[,<seqlen>][,<cmdseq>]

<seqlen> ist dabei eine Dezimalzahl, die angibt, wieviele Bytes zusätzlich zu den standardmäßigen 512 Bytes für den Sequenz - Editfile zur Verfügung gestellt werden sollen. Will man das Copy - Kommando auf den Sequenz - Editfile anwenden, so muß hier eine Zahl größer Null angegeben werden, da sonst der bei Copy intern benötigte Speicherplatz nicht ausreicht.

<cmdseq> ist eine beliebige Kommandosequenz, die EDI direkt nach dem Start ausführt. Beispiel:

EDI,200,E,TEST/MAC!L

ruft EDI auf, reserviert $512 + 200 = 712$ Bytes für den Sequenz - Editfile, liest den File TEST/MAC von Diskette ein und listet die ersten 15 Zeilen.

6.5 PARAMETER

Eine Kommandosequenz kann Parameter enthalten, die erst beim Aufruf eingesetzt werden müssen. Diese Parameter haben die Form #n, wobei n die Nummer des Parameters ist. Man sollte

diese Nummern von 1 an lückenlos aufsteigend verteilen. Die Ersetzung dieser Formalparameter geschieht beim Aufruf durch das Q-Kommando, z.B.:

Q,1=L,#1,#2/	Abspeichern der Sequenz
Q,1,A,HUGO	Erzeugt das Kommando L,A,/HUGO/, indem A für #1 und HUGO für #2 gesetzt wird.

Die Zahl der Parameter ist beliebig, allerdings darf ein Kommando nicht länger als 255 Zeichen werden. Andernfalls meldet EDI: COMMAND TOO LONG.

6.6 KOMMANDOSEQUENZEN AUF DISKETTE

Über die normalen Edit - und Save - Kommandos können Sie auch Kommandosequenzen auf Diskette abspeichern und wieder einlesen. Beispiel:

```
S,<filename>,Q
E,<filename>,Q
```

Diese Kommandos speichern den gesamten Sequenz - Edit-file ab und lesen ihn wieder ein. Lesen Sie die Kapitel 3.8 und 3.9 für die genaue Erklärung der weiteren möglichen Parameter für diese beiden Kommandos.

Auf der Diskette, auf der Sie EDI bekommen haben, sind auch zwei Files FORTRAN/EDI und MACRO/EDI enthalten. Sie enthalten Kommandosequenzen, die für FORTRAN - Programme bzw. Assembler - Programme besonders nützlich sein können. Eine Erklärung dieser Sequenzen finden Sie in Kapitel 7.

6.7 BEISPIELE

Das folgende Beispiel ist eine Kommandosequenz, die alle PROGRAM-, SUBROUTINE- oder FUNCTION-Zeilen aus einem FORTRAN - Programm listet und zwar in der Reihenfolge, in der sie im Programm vorkommen.

```
Q,1=V5=0!
Q,2=V1=$5+1!Q,7,PROGRAM,2!Q,7,SUBROUTINE,3!
Q,3=Q,7,FUNCTION,4!
Q,4=V5=$1!W,+, $2,V1=$2!Q,6,3!Q,6,4!
Q,5=W,-Z,$1-$5,L,$5!W,-Z,$1-$5,Q,2
Q,6=W,+, $#1,W,-, $1-$#1,V1=$#1
Q,7=F,$1,L,/ #1/!,V#2=*
```

In Zeile 1 wird die Variable V1, die als Zeilenzeiger benutzt wird, auf Null gesetzt. Zeile 2 erhöht den Zeiger und sucht nach einer Zeile, die das Wort PROGRAM enthält. Die Nummer dieser Zeile kommt dann nach V2. Gibt es keine solche Zeile, so wird V2 auf -1 gesetzt. Dasselbe wird nun für SUBROUTINE und in Zeile 3 für FUNCTION getan und die Ergebnisse in den Variablen V3 und V4 abgespeichert. In Zeile 4 wird das Minimum von V2, V3 und V4 gesucht und in V5

gespeichert. Zeile 5 listet nun die entsprechende Zeile auf dem Bildschirm und fängt bei Zeile 2 wieder an, die nächste Zeile zu suchen. In der Zeile 6 wird ein einzelner Vergleich durchgeführt, der zur Berechnung des Minimums benötigt wird. Zeile 7 sucht nach einem Wort und speichert die gefundene Zeilennummer in einer Variablen ab.

6.8 BESCHRÄNKUNGEN

Kommandosequenzen unterliegen einigen typischen Beschränkungen, die beachtet werden sollten. Diese sind:

- Ein Kommando in einer Sequenz darf kein Ausrufezeichen enthalten, auch nicht in einem Suchstring. Das Ausrufezeichen wird immer als Trennzeichen zwischen den einzelnen Kommandos interpretiert.
- Ein Kommando in einer Sequenz darf nicht das Zeichen # enthalten, da dieses Zeichen für formale Parameter vorbehalten ist.
- Ein Parameter des Q-Kommandos kann kein Komma enthalten, da Kommata die einzelnen Parameter voneinander trennen.
- In einem Q-Kommando dürfen nur die ersten beiden Kommata entfallen. Alle weiteren sind zur Trennung der Formalparameter erforderlich.

7. MITGELIEFERTE KOMMANDOSEQUENZEN

=====

7.1 FORTRAN/EDI

Der File FORTRAN/EDI enthält Kommandosequenzen, die besonders für die Programmierung in FORTRAN nützlich sind. Lesen Sie die Sequenzen von Diskette ein durch die Kommandos:

D, @, A
E, FORTRAN/EDI, @

Die eingelesenen Sequenzen führen folgende Aktionen durch:

- Q,1 Fügt vor jede PROGRAM-, SUBROUTINE-, FUNCTION- oder BLOCK DATA - Zeile eine leere Kommentarzeile ein, die nur ein C in Spalte eins enthält. Dadurch werden die einzelnen Programmteile übersichtlich voneinander getrennt.
- Q,2 Löscht alle Kommentarzeilen, die nur ein C in Spalte eins und keinen Text enthalten. Solche Zeilen werden durch Q,1 erzeugt.
- Q,3 Löscht alle Kommentarzeilen.
- Q,4 Listet alle Zeilen mit Statementnummer.
- Q,5 Listet alle DO - Statements. EDI sucht dabei nach Zeilen, die sowohl das Wort DO als auch ein Komma und ein Gleichheitszeichen enthalten.

Sehen Sie sich die Sequenzen genau an. Sie können daraus auch bei der Erstellung eigener Sequenzen Nutzen ziehen. Falls Sie nicht gleich klarkommen, setzen Sie doch einmal die weggelassenen Kommata in die Sequenzen ein.

7.2 MACRO/EDI

Dieser File enthält Sequenzen, die Sie bei der Erstellung von Assembler - Programmen benutzen können. Einige der Kommandosequenzen können aber auch in anderem Zusammenhang nützlich sein. Laden Sie den File durch die Kommandos:

D, @, A
E, MACRO/EDI, @

Die einzelnen Sequenzen führen folgende Funktionen aus:

- Q,1 Formatiert den Editfile so, daß alle Opcodes in Spalte 10 beginnen, alle Operanden in Spalte 17 und Kommentare in Spalte 32.

- Q,2 Formatiert den Editfile so, daß er möglichst wenig Platz benötigt. Überflüssige Blanks werden entfernt. Beachten Sie, daß durch diese Sequenz auch alle Textkonstanten geändert werden, die mehr als zwei Leerzeichen hintereinander enthalten. Dies kann unter Umständen zu fehlerhaften Ergebnissen führen.
- Q,3 Listet alle Zeilen mit Label.
- Q,4 Merkt sich die Nummer der aktuellen Zeile in der Variablen V9.
- Q,5 Listet den Bereich um die Zeile, die mit Q,4 festgehalten wurde.

Sie können selbst Erweiterungen zu den genannten Sequenzen machen. Dabei müssen Sie jedoch unter Umständen die Größe des Sequenz - Editfiles erhöhen, indem Sie beim Start von EDI die Zahl der zusätzlichen Bytes angeben, z.B. mit
EDI,300

Dadurch werden 300 Bytes zusätzlich für den Sequenz - Editfile bereitgestellt, insgesamt also 812 Bytes.

Anhang A ZUSAMMENFASSUNG DER KOMMANDOS

A[,<bereich>][,<suchstr>]

Ändere Zeilen.

B

Verlasse EDI und gehe ins DOS.

C,<ausdruck>[,<bereich>][,<suchstr>][,<V>]

Kopiere Zeilen innerhalb des Editfiles.

D[,<bereich>][,<suchstr>][,<V>]

Lösche Zeilen.

E,<filename>[,<ausdruck>]

Lies einen File von Diskette.

FC,<bereich>][,<suchstr>]

Suche angegebene Zeile.

I[,<ausdruck>]

Zeilen eingeben.

J,<spalte>[,<bereich>][,<suchstr>]

Mit Blanks auffüllen oder Blanks abschneiden.

L[,<bereich>][,<suchstr>]

Zeilen auf dem Bildschirm listen.

M,<ausdruck>[,<bereich>][,<suchstr>][,<V>]

Bewegen von Zeilen innerhalb des Editfiles.

N

Spaltennummern feststellen.

PC,<bereich>][,<suchstr>]

Zeilen auf den Drucker ausgeben.

Q,<ausdruck>[,<parameter>]...

Kommandosequenz ausführen.

S[+/-],<filename>[,<bereich>][,<suchstr>]

Editfile auf Diskette schreiben.

T[,<spalte>]...

Setzen und Abfragen von Tabulatorstops.

V<ziffer19>[=<ausdruck>]

Variable setzen oder abfragen.

W,<bed>,<ausdruck>,<edicmd>

Bedingte Ausführung eines Kommandos.

X,<doscmd>

DOS - Kommando ausführen (nur NEWDOS80).

<suchstr1>=<string>[,<bereich>][,<suchstr2>][,<V>]

String ersetzen.

<ausdruck>=<text>

Eingeben einzelner Zeilen.

Anhang B ZUSAMMENFASSUNG DER NOTATION

$\langle \text{item} \rangle = F / L / * / \$\langle \text{ziffer19} \rangle / \langle \text{zahl} \rangle$
 $\langle \text{ziffer19} \rangle = 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9$
 $\langle \text{ziffer} \rangle = \langle \text{ziffer19} \rangle / 0$
 $\langle \text{zahl} \rangle = \langle \text{ziffer} \rangle \dots$
 $\langle \text{ausdruck} \rangle = [+ / -] \langle \text{item} \rangle [(+ / -) \langle \text{item} \rangle] \dots$
 $\langle \text{bereich} \rangle = A / \langle \text{ausdruck} \rangle [, \langle \text{ausdruck} \rangle]$
 $\langle \text{suchstr} \rangle = \langle \text{string} \rangle [, \langle \text{spalte} \rangle [, \langle \text{spalte} \rangle]] [, U]$
 $\langle \text{string} \rangle = / \langle \text{text} \rangle /$
 $\langle \text{spalte} \rangle = \langle \text{ausdruck} \rangle , \text{ der nicht F, L oder * enthält}$
 $\langle \text{text} \rangle = \langle \text{zeichen} \rangle \dots$
 $\langle \text{edicmd} \rangle = \text{irgend ein gültiges EDI - Kommando}$
 $\langle \text{doscmd} \rangle = \text{irgend ein gültiges DOS - Kommando}$
 $\langle \text{filename} \rangle = \text{ein gültiger DOS - Filename}$
 $\langle \text{bed} \rangle = + / - / Z / -Z$

Anhang C FEHLERMELDUNGEN

ILLEGAL COMMAND

EDI konnte das eingegebene Kommando nicht als gültig identifizieren.

ILLEGAL TEXT FILE

Eine Zeile des Files, der mit dem E-Kommando eingelesen werden sollte, war länger als 137 Zeichen. Der File ist deshalb nicht von EDI bearbeitbar.

MEMORY FULL

Das eingegebene Kommando hätte mehr Speicherplatz benötigt, als zur Verfügung stand. Es ist so weit wie möglich ausgeführt worden. (Move oder Copy werden gar nicht ausgeführt, wenn der Speicherplatz nicht ausreicht.) Auch eine zu hohe Schachteltiefe bei Sequenz - Aufrufen kann zu diesem Fehler führen.

LINE TRUNCATED

Eine Zeile wäre länger als 137 Zeichen geworden und muß deshalb abgeschnitten werden. Passiert das beim String - Ersetze - Kommando, dann geht EDI automatisch in den Veto - Modus über und überläßt dem Benutzer die Entscheidung, ob die Zeile abgeschnitten werden soll oder ob die Änderungen an dieser Zeile nicht durchgeführt werden sollen.

ILLEGAL IN SEQ-MODE

Kommandos, die den Sequenz - Editfile verändern, dürfen in Kommandosequenzen nicht verwendet werden, außer als letztes Sequenz - Kommando, bevor EDI wieder Eingaben von der Tastatur annimmt.

COMMAND TOO LONG

Durch die Ersetzung von Parametern beim Aufruf einer Kommandosequenz wurde ein EDI - Kommando länger als 255 Zeichen. Dies ist nicht zulässig.

PARAMETER MISSING

Beim Q-Kommando fehlte mindestens ein Parameter. Wenn nicht der Systemparameter SUPPRESS SEQ ECHO auf Y steht, dann zeigt EDI das betreffende Kommando bis zu der Stelle an, an der der fehlende Parameter benötigt worden wäre.

NO SUCH LINE

Informative Meldung die besagt, daß keine Zeile den angegebenen Spezifikationen entsprach. Während der Ausführung einer Kommandosequenz wird diese Meldung unterdrückt.

ILLEGAL STRING

Ein String war nicht ordnungsgemäß mit einem Schrägstrich abgeschlossen.

ILLEGAL NUMBER

Für diese Fehlermeldung kommen folgende Ursachen in Frage: Eine Zeilennummer war zu groß oder Null oder negativ. Eine Spaltennummer war größer als 137 oder Null oder negativ. Es wurde beim Ansprechen einer Variablen keine der Ziffern 1 bis 9 benutzt. Es sollte ein Tabulator über Spalte 58 hinaus gesetzt werden.

PRINTER NOT READY

Der Drucker ist nicht bereit, das eingegebene P-Kommando auszuführen. Es ist entweder kein Papier eingelegt, der Drucker ist ausgeschaltet oder es ist überhaupt kein Drucker angeschlossen. Diese Fehlermeldung kann im System Configuration Mode (s. 4.7) erlaubt oder unterdrückt werden.

COMMAND ABORTED

Wenn ein Kommando mittels der Clear - Taste abgebrochen wird, dann erscheint diese Meldung.

ILLEGAL EXPRESSION

Diese Meldung zeigt an, daß ein Ausdruck nicht die erwartete Form hatte.

Zusätzlich können alle ODS - Fehlermeldungen auftreten, wie z.B.

FILE NOT IN DIRECTORY

DISKETTE SPACE FULL

DEVICE NOT AVAILABLE

usw.